



(19)

(11) Publication number: **06131218 A**

Generated Document.

**PATENT ABSTRACTS OF JAPAN**(21) Application number: **04277453**(51) Int. Cl.: **G06F 11/30 G06F 1/24**(22) Application date: **15.10.92**

(30) Priority:

(43) Date of application publication: **13.05.94**

(84) Designated contracting states:

(71) Applicant: **FUJITSU LTD**(72) Inventor: **ISHIKAWA HITOSHI**

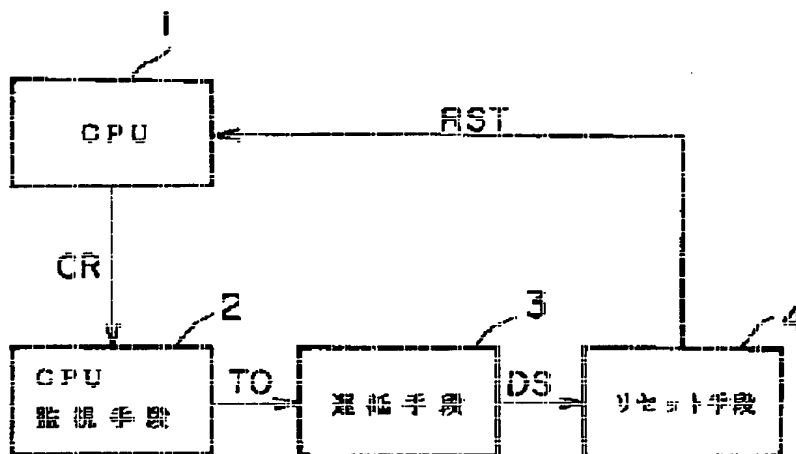
(74) Representative:

**(54) CPU CONTROL SYSTEM**

(57) Abstract:

**PURPOSE:** To surely restart a computer system by the CPU control system even in case of the destruction of a control table, etc., with respect to a CPU control system.

**CONSTITUTION:** A CPU monitoring means 2 outputs a time-over signal TO unless a counter reset signal CR is inputted within a specific cycle period. A delay means 3 on receiving the time-over signal TO outputs a delay signal DS which is delayed by a specific time. A resetting means 4 once inputting delay signal DS outputs a reset signal RST. Therefore, the CPU 1 is reset with the delay signal DS unless an NMI(Non-Maskable Interrupt) processing program which is started when the time-over signal is inputted as an NMI signal to the CPU 1 is started since the control table, etc., is destroyed, so the computer system can surely be restarted.



COPYRIGHT: (C)1994,JPO&amp;Japio

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平6-131218

(43)公開日 平成 6 年(1994) 5 月13日

(51)Int.Cl.<sup>5</sup>

G 0 6 F 11/30  
1/24

識別記号

3 1 0 K

庁内整理番号

9290-5B

7165-5B

F I

G 0 6 F 1/ 00

技術表示箇所

3 5 0 B

審査請求 未請求 請求項の数 4 (全 7 頁)

(21)出願番号

特願平4-277453

(22)出願日

平成 4 年(1992)10月15日

(71)出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中1015番地

(72)発明者 石川 均

神奈川県川崎市中原区上小田中1015番地

富士通株式会社内

(74)代理人 弁理士 井桁 貞一

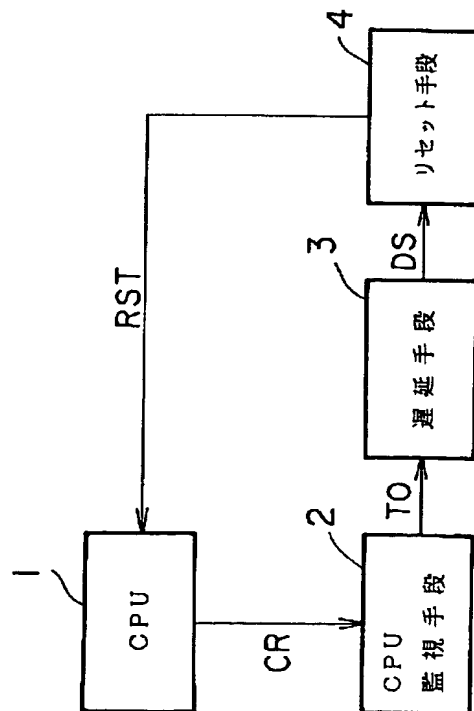
(54)【発明の名称】 CPU制御方式

(57)【要約】

【目的】 CPU制御方式に関し、管理テーブル等が破壊された場合であっても、確実に計算機システムの再起動が行えることを目的とする。

【構成】 カウンタリセット信号CRが所定の周期期間内に入力されなかった場合、CPU監視手段2はタイムオーバー信号TOを出力する。タイムオーバー信号TOを受けた遅延手段3は、所定の時間遅らせた遅延信号DSを出力する。リセット手段4は遅延信号DSを入力すると、リセット信号RSTを出力する。したがって、タイムオーバー信号TOがNMI (Non-Maskable Interrupt) 信号としてCPU1に入力されると起動されるNMI処理プログラムが、管理テーブル等が破壊されたために起動されない場合は遅延信号DSにより、CPU1をリセットするので、確実に計算機システムの再起動を行うことができる。

本発明の原理説明図



## 1

## 【特許請求の範囲】

【請求項1】 計算機システムの異常を検出してプロセッサ（CPU）をリセットするCPU制御方式において、

CPU（1）から周期的に出力されるカウンタリセット信号（CR）を監視し、前記カウンタリセット信号（CR）が所定の周期期間内に入力されなかった場合はタイムオーバー信号（TO）を出力するCPU監視手段

（2）と、

前記タイムオーバー信号（TO）を入力し、所定の時間遅らせた遅延信号（DS）を出力する遅延手段（3）と、

前記遅延信号（DS）を入力した場合には、前記CPU（1）をリセットするためのリセット信号（RST）を出力するリセット手段（4）と、

を有することを特徴とするCPU制御方式。

【請求項2】 前記CPU監視手段（2）は、ウォッチドグタイマー（Watch Dog Timer）で構成したことを特徴とする請求項1記載のCPU制御方式。

【請求項3】 前記遅延手段（3）と前記リセット手段（4）との間に設けられ、前記CPU（1）から出力禁止信号を入力した場合には、前記遅延信号（DS）の出力を禁止する信号出力制御手段を、さらに有することを特徴とする請求項1記載のCPU制御方式。

【請求項4】 前記遅延手段（3）に設定する前記所定の時間は、NMI（Non-Maskable Interrupt）処理プログラムの実行時間よりも長い時間であるように構成したことを特徴とする請求項1記載のCPU制御方式。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】 本発明はCPU制御方式に関し、特に計算機システムの異常を検出してプロセッサ（CPU）をリセットするCPU制御方式に関する。

【0002】 近年、情報通信機器のみならず家電製品や自動車等の様々な工業製品に対して、コンピュータすなわちプロセッサ（以下、単に「CPU」と呼ぶ。）が組み込まれ、一種の計算機システムを構成している。これらの計算機システムは、それぞれの工業製品に応じて目的の動作が機能するように、様々な信号の制御等を行っている。

【0003】 一般に、何らかの障害の発生によりシステムダウンを防止するため、計算機システムには様々なフェールセーフ（fail safe）機能が設けられている。こうしたフェールセーフ機能の一つに、システムの誤りを検出する手段として、ウォッチドグタイマー（Watch Dog Timer）がある。

【0004】 このウォッチドグタイマーは計算機システムの状態又は所定の信号を監視し、所定の時間を超えて変化しない場合、計算機システムに誤りが生じたものとして検出するタイマーである。したがって、ウォッチド

## 2

グタイマーに対して一定周期ごとに上記所定の信号としてのパルスを与えるように構成しておく必要がある。例えば、ソフトウェアの内部に上記パルスを発生させるための命令を組み込み、この命令を一定周期ごとに実行するような構成をとる。もし、ソフトウェアの無限ループやハードウェア内部のフォールト（fault）等が発生すると、一定周期以内にパルスを発生させることができなくなる。その結果、ウォッチドグタイマーは計算機システムに誤りが生じたものと判断する。

【0005】 システムの誤りが検出されると、ウォッチドグタイマーはタイムオーバー信号をCPUへ出力する。このタイムオーバー信号は割込信号の一つであるNMI（Non-Maskable Interrupt）信号としてCPUに入力され、自動的にシステムを初期化して再起動を行う。こうして、計算機システムのシステムダウンを防止している。

## 【0006】

【従来の技術】 図4は、ウォッチドグタイマーを用いた計算機システムのフェールセーフ機能を実現する回路構成であって、従来の回路構成を示すブロック図である。

【0007】 図において、フェールセーフ機能を実現する回路は、CPU21、WDT（ウォッチドグタイマー）回路22、フェッチレジスタ23、カウンタリセットレジスタ24、システムリセットレジスタ25及びシステムリセット回路26から構成される。なお、CPU21、フェッチレジスタ23、カウンタリセットレジスタ24及びシステムリセットレジスタ25は、バス29に互いに接続されている。

【0008】 CPU21は図示されていないROMに格納されたシステムプログラムを実行することによって、計算機システム全体の制御を行う。WDT回路22は内部にカウンタ及びタイマーを有し、このタイマーが所定時間経過するごとにカウンタを1だけ増加させる。カウンタは外部からカウンタリセット信号CRが入力されるとゼロに初期化され、所定のカウンタ値に達するとタイムオーバー信号TOを出力する。なお、WDT回路22は後述するシステムリセット回路26からのリセット信号RSTを入力すると、タイムオーバー信号TOの出力を停止する。

【0009】 フェッチレジスタ23は上記タイムオーバー信号TO等のハードウェアで生じたエラー信号を検出し、発生したハードウェアの種別等を、バス29を介してCPU21へ通知する。カウンタリセットレジスタ24はバス29を介してなされるCPU21からのカウンタリセット指令によって、WDT回路22へカウンタリセット信号CRを出力する。システムリセットレジスタ25はバス29を介してなされるCPU21からのシステムリセット指令によって、システムリセット回路26へシステムリセット信号SR1を出力する。

【0010】 システムリセット回路26はシステムリセ

## 3

ット信号SR1又はパワースイッチSWのうちいずれかを検出して、CPU21及びWDT回路22へリセット信号RSTを出力する。

【0011】次に、従来のフェールセーフ機能を実現する回路の動作について説明する。CPU21は所定の周期ごとにカウンタリセット指令を、バス29を介してカウンタリセットレジスタ24に行う。カウンタリセット指令を受けたカウンタリセットレジスタ24は、WDT回路22へカウンタリセット信号CRを出力する。

【0012】一方、WDT回路22の内部ではタイマーが駆動しており、このタイマーが所定時間経過するごとにカウンタを1だけ増加させる。このカウンタはカウンタリセット信号CRを入力するとゼロに初期化され、所定のカウンタ値に達するとタイムオーバー信号TOを出力する。したがって、所定の周期期間内にカウンタリセット信号CRが入力されなければ、タイムオーバー信号TOを出力する。

【0013】このタイムオーバー信号TOはNMI信号としてCPU21に入力され、NMI処理プログラムが起動される。一般に、NMI処理プログラムはシステムの再起動を行うためにシステムリセット指令を、バス29を介してシステムリセットレジスタ25に行う。システムリセット指令を受けたシステムリセットレジスタ25は、システムリセット回路26へシステムリセット信号SR1を出力する。そして、システムリセット信号SR1を受けたシステムリセット回路26は、CPU21及びWDT回路22へリセット信号RSTを出力する。こうして、CPU21はシステムの再起動を行うことができる。

【0014】なお、上記NMI処理プログラムは、必要に応じてフェッチレジスタ23からエラー信号発生の原因となったハードウェアの種別等を取得し、メモリ内容の退避等の適切なエラー回復措置を行なった後にシステムリセット指令を行い、計算機システムを再起動する。

【0015】次に、上記計算機システムにおけるメモリ配置について説明する。図5は、メモリ配置の一例を示す図である。図において、メモリ100には最低位アドレスから最高位アドレスに向かって、順に割込ベクタ(Interrupt Vector)領域101、オペレーティングシステム(Operating System; 以下、単に「OS」と呼ぶ。)領域102、アプリケーションプログラム(Application Program; 以下、単に「AP」と呼ぶ。)領域103、初期化プログラム領域104及びジャンプ命令領域105が割り当てられ、それぞれの領域には対応するアドレスデータ又はプログラム等が格納されている。

【0016】割込ベクタ領域101にはカウンタリセット指令を行うプログラム又はNMI処理プログラム等の割込プログラムが格納されているアドレスが設定されている。CPU21は割り込み要求があると、割込ベクタ領域101から割込番号に対応する割込プログラムが格

## 4

納されているアドレスを取得する。その後、CPU21はこのアドレスへジャンプして該当するプログラムを実行することにより、割込処理を実現する。

【0017】OS領域102にはMS-DOS(登録商標)等のOSが格納され、管理テーブル102aを基にプロセス管理、記憶管理、入出力制御及びデータ管理等の各種管理の処理を行う。AP領域103には、一般にユーザー等が作成した実行プログラムが格納される。初期化プログラム領域104には計算機システム全体又は一部を使用可能に初期化するためのプログラムが格納される。ジャンプ命令領域105にはCPU21をリセットした際に、最初に行われる命令が格納される。このジャンプ命令領域105は、一般に初期化プログラム領域104へジャンプする命令が格納される。

【0018】このようなメモリ配置によって、ウォッチドグタイマーが計算機システムに誤りが発生したことを検出し、CPU21がNMI信号を受けると割込ベクタ領域101に格納されたアドレスを参照してNMI処理プログラムを実行する。さらにはシステムリセット回路26からのリセット信号RSTを受けてジャンプ命令領域105に格納された命令及び初期化プログラム領域104に格納された初期化プログラムを実行して、計算機システムの再起動を実現している。

## 【0019】

【発明が解決しようとする課題】しかし、AP領域103に格納されたプログラムが、例えば計算機システム外部からのノイズやアプリケーションプログラムのミス等によってアドレスデータ又はプログラム等が破壊されたために暴走し、割込ベクタ領域101及び管理テーブル102aに格納されたアドレスを破壊する場合がある。このような場合は計算機システムの再起動ができず、システムダウンしてしまうという問題点があった。

【0020】本発明はこのような点に鑑みてなされたものであり、管理テーブル等が破壊された場合であっても、確実に計算機システムの再起動が行えるCPU制御方式を、提供することを目的とする。

## 【0021】

【課題を解決するための手段】図1は上記目的を達成する本発明の原理説明図である。図において、本発明のCPU制御方式はプロセッサ(CPU)監視手段2、遅延手段3及びリセット手段4から構成される。

【0022】CPU監視手段2はCPU1から周期的に出力されるカウンタリセット信号CRを監視し、カウンタリセット信号CRが所定の周期期間内に入力されなかった場合はタイムオーバー信号TOを出力する。遅延手段3はタイムオーバー信号TOを入力し、所定の時間遅らせた遅延信号DSを出力する。リセット手段4は遅延信号DSを入力した場合には、CPU1をリセットするためのリセット信号RSTを出力する。

## 【0023】

## 5

【作用】カウンタリセット信号CRが所定の周期期間内に入力されなかった場合、CPU監視手段2はタイムオーバー信号TOを出力する。タイムオーバー信号TOを受けた遅延手段3は、所定の時間遅らせた遅延信号DSを出力する。リセット手段4は遅延信号DSを入力すると、CPU1をリセットするためのリセット信号RSTを出力する。

## 【0024】

【実施例】以下、本発明の一実施例を図面に基いて説明する。図2は、本発明の実施例を示すブロック図である。図において、本発明のCPU制御方式は、CPU11、WDT（ウォッチドグタイマー）回路12、フェッチレジスタ13、カウンタリセットレジスタ14、システムリセットレジスタ15、システムリセット回路16及びディレイ回路17から構成される。なお、CPU11、フェッチレジスタ13、カウンタリセットレジスタ14及びシステムリセットレジスタ15は、バス19に互いに接続されている。

【0025】CPU11は図示されていないROMに格納されたシステムプログラムを実行することによって、計算機システム全体の制御を行う。WDT回路12は内部にカウンタ及びタイマーを有し、このタイマーが所定時間経過するごとにカウンタを1だけ増加させる。カウンタは外部からカウンタリセット信号CRが入力されるとゼロに初期化され、所定のカウンタ値に達するとタイムオーバー信号TOを出力する。なお、WDT回路12は後述するシステムリセット回路16からのリセット信号RSTを入力すると、タイムオーバー信号TOの出力を停止する。

【0026】フェッチレジスタ13は上記タイムオーバー信号TO等のハードウェアで生じたエラー信号を検出し、発生したハードウェアの種別等を、バス19を介してCPU11へ通知する。カウンタリセットレジスタ14はバス19を介してなされるCPU11からのカウンタリセット指令によって、WDT回路12へカウンタリセット信号CRを出力する。システムリセットレジスタ15はバス19を介してなされるCPU11からのシステムリセット指令によって、システムリセット回路16へシステムリセット信号SR1を出力する。なお、これらのフェッチレジスタ13、カウンタリセットレジスタ14及びシステムリセットレジスタ15は、例えばゲート回路で構成される。

【0027】ディレイ回路17は例えば積分回路によって構成され、WDT回路12から出力されたタイムオーバー信号TOを入力し、所定の時間だけ遅延させ、遅延信号DSとしてシステムリセット回路16へ出力する。ここで、遅延させる所定の時間は、NMI処理プログラムをCPU11が実行する時間よりも長い時間に設定される。システムリセット回路16は遅延信号DS、システムリセット信号SR1又はパワースイッチSWのうち

## 6

いずれかを検出して、CPU11及びWDT回路12へリセット信号RSTを出力する。

【0028】次に、上記実施例の動作について説明する。CPU11は所定の周期ごとにカウンタリセット指令を、バス19を介してカウンタリセットレジスタ14に行う。カウンタリセット指令を受けたカウンタリセットレジスタ14は、WDT回路12へカウンタリセット信号CRを出力する。

【0029】一方、WDT回路12の内部ではタイマーが駆動しており、このタイマーが所定時間経過するごとにカウンタを1だけ増加させる。このカウンタはカウンタリセット信号CRを入力するとゼロに初期化され、所定のカウンタ値に達するとタイムオーバー信号TOを出力する。したがって、所定の周期期間内にカウンタリセット信号CRが入力されなければ、タイムオーバー信号TOを出力する。

【0030】このタイムオーバー信号TOはNMI信号としてCPU11に入力され、NMI処理プログラムが起動される。一般に、NMI処理プログラムはシステムの再起動を行うためにシステムリセット指令を、バス19を介してシステムリセットレジスタ15に行う。また、必要に応じてフェッチレジスタ13からエラー信号発生の原因となったハードウェアの種別等を取得し、メモリ内容の退避等の適切な処置を行なった後にシステムリセット指令を行う。そして、システムリセット指令を受けたシステムリセットレジスタ15は、システムリセット回路16へシステムリセット信号SR1を出力する。

【0031】また、タイムオーバー信号TOはディレイ回路17へ入力され、所定の時間だけ遅延させて遅延信号DSとしてシステムリセット回路16へ出力する。さらに、システムリセット信号SR1又は遅延信号DSを受けたシステムリセット回路16は、CPU11及びWDT回路12へリセット信号RSTを出力する。

【0032】したがって、NMI処理プログラムが正常に起動された場合はシステムリセット信号SR1により、図5に示す割込ベクタ領域101及び管理テーブル102aに格納されたアドレスが破壊されたためにNMI処理プログラムが起動されない場合は遅延信号DSにより、確実にCPU11をリセットすることができ、計算機システムの再起動を行うことができる。このため、本発明を備えた計算機システムの信頼性が高まる。

【0033】また、ディレイ回路17はタイムオーバー信号TOを入力して遅延信号DSとして出力するまでの時間を、NMI処理プログラムをCPU11が実行する時間よりも長くするように構成したので、NMI処理プログラムの実行中にCPU11がリセットされることがなくなり、確実にメモリ内容の退避等の適切なエラー回復措置を行うことができる。

【0034】図3は、本発明の他の実施例を示すブロッ

ク図である。なお、図2と同一の要素には同一番号を付し、説明を省略する。図において、図2に示す本発明の実施例と異なる点は、バス19に接続されたゲート制御レジスタ18aと、ディレイ回路17から出力される遅延信号DSとゲート制御レジスタ18aから出力される出力禁止信号ESとの論理積をとる論理積回路18bとが設けられたことである。

【0035】ゲート制御レジスタ18aは、NMI処理プログラムをCPU11が実行することによって指令される出力禁止指令を受けて、出力禁止信号ESを出力する。具体的には、通常ゲート制御レジスタ18aは「1」の出力許可信号を出力し、出力禁止指令を受けた後は出力許可指令を受けるか、又はCPU11がリセットされるまで「0」の出力禁止信号ESを出力する。

【0036】論理積回路18bは遅延信号DSの出力を制御する信号出力制御手段であって、ゲート制御レジスタ18aから「1」の出力許可信号を受けると遅延信号DSをシステムリセット信号SR2として出力し、ゲート制御レジスタ18aから「0」の出力禁止指令ESを受けると遅延信号DSの出力を停止する。

【0037】したがって、NMI処理プログラムをCPU11が実行するために要する時間が、予め設定されたディレイ回路17の所定の時間を超えるような場合、NMI処理プログラムの中に遅延信号DSの出力を禁止する出力禁止指令を組み込むことによって遅延信号DSの出力を制御することができる。このため、NMI処理プログラムの実行中にディレイ回路17に設定された所定の時間を超えた場合であってもCPU11がリセットされることがなくなり、確実にメモリ内容の退避等の適切なエラー回復措置を行うことができる。

【0038】上記の説明では、ディレイ回路17は積分回路等によって構成して所定の時間だけ遅延させた信号を出力したが、ディレイ回路17をタイマー及びカウンタを含み、CPU11からバス19を介してカウンタの

カウンタ値を設定可能にロジックで構成してもよい。例えば、CPU11からの指令によって設定されたカウンタをタイマーが所定時間経過するごとにカウンタを1だけ減算し、カウンタがゼロに達したときにタイムオーバー信号TOを出力するように構成する。こうすることによって、NMI処理プログラムの実行に必要な時間をCPU11から直接ディレイ回路17のカウンタに設定することができるので、ゲート制御レジスタ18a等を設けることなく、確実にメモリ内容の退避等の適切なエラー回復措置を行うことができる。

【0039】また、フェッチレジスタ13及びカウンタリセットレジスタ14等のレジスタはゲート回路によって構成したが、1個のレジスタに対して1ビットのメモリを割り当てるように構成してもよい。

【0040】

【発明の効果】以上説明したように本発明では、CPU監視手段から出力されたタイムオーバー信号を受けた遅延手段が所定の時間遅らせた遅延信号を出力し、リセット手段がCPUをリセットするためのリセット信号を出力するように構成したので、管理テーブル等が破壊された場合であっても、確実に計算機システムの再起動を行うことができる。

【図面の簡単な説明】

【図1】本発明の原理説明図である。

【図2】本発明の実施例を示すブロック図である。

【図3】本発明の他の実施例を示すブロック図である。

【図4】従来の回路構成を示すブロック図である。

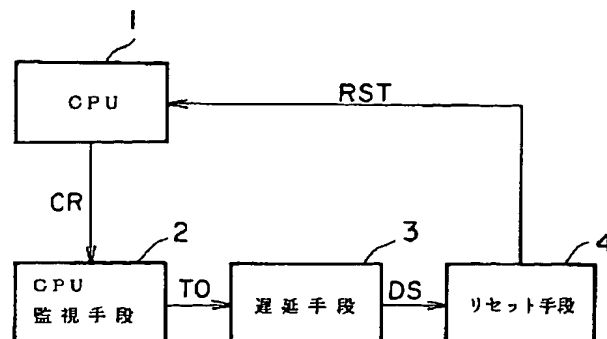
【図5】メモリ配置の一例を示す図である。

【符号の説明】

- 1 プロセッサ (CPU)
- 2 プロセッサ (CPU) 監視手段
- 3 遅延手段
- 4 リセット手段

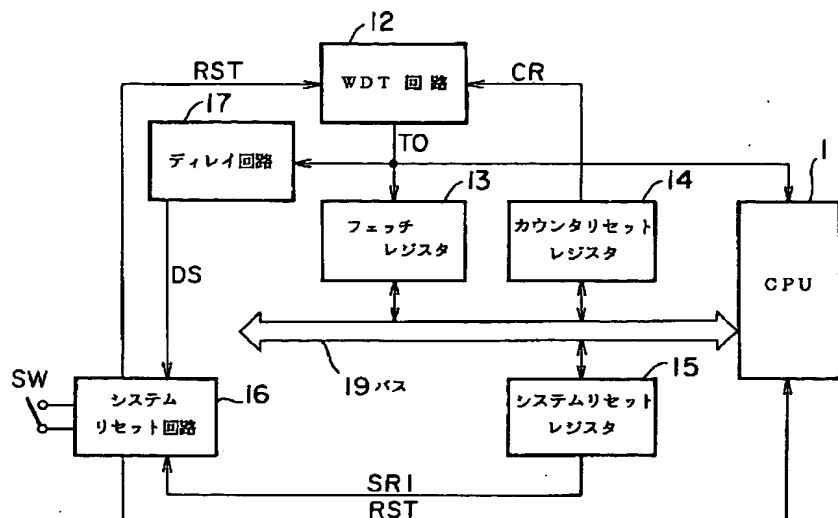
【図1】

本発明の原理説明図



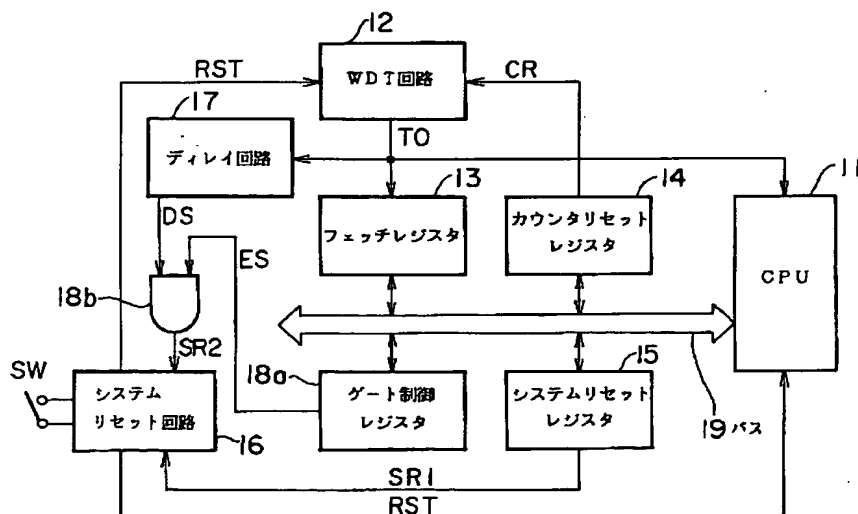
【図2】

本発明の実施例を示すブロック図



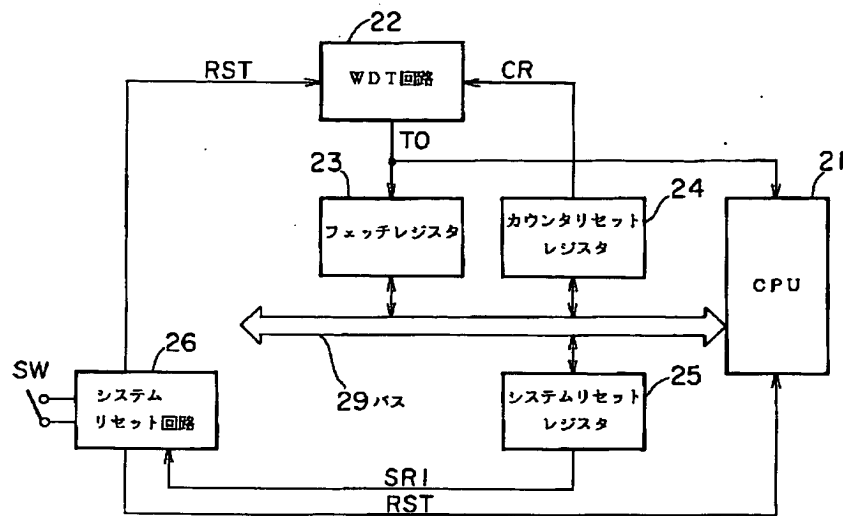
【図3】

本発明の他の実施例を示すブロック図



【図4】

従来の回路構成を示すブロック図



【図5】

メモリ配置の一例を示す図

